*A Secure Bandwidth Reservation System**

Gary Hoo, Keith Jackson and William Johnston, Lawrence Berkeley National Laboratory

Ian Foster and Alain Roy, Argonne National Laboratory and University of Chicago

Volker Sander, Argonne National Laboratory

HPDC '99

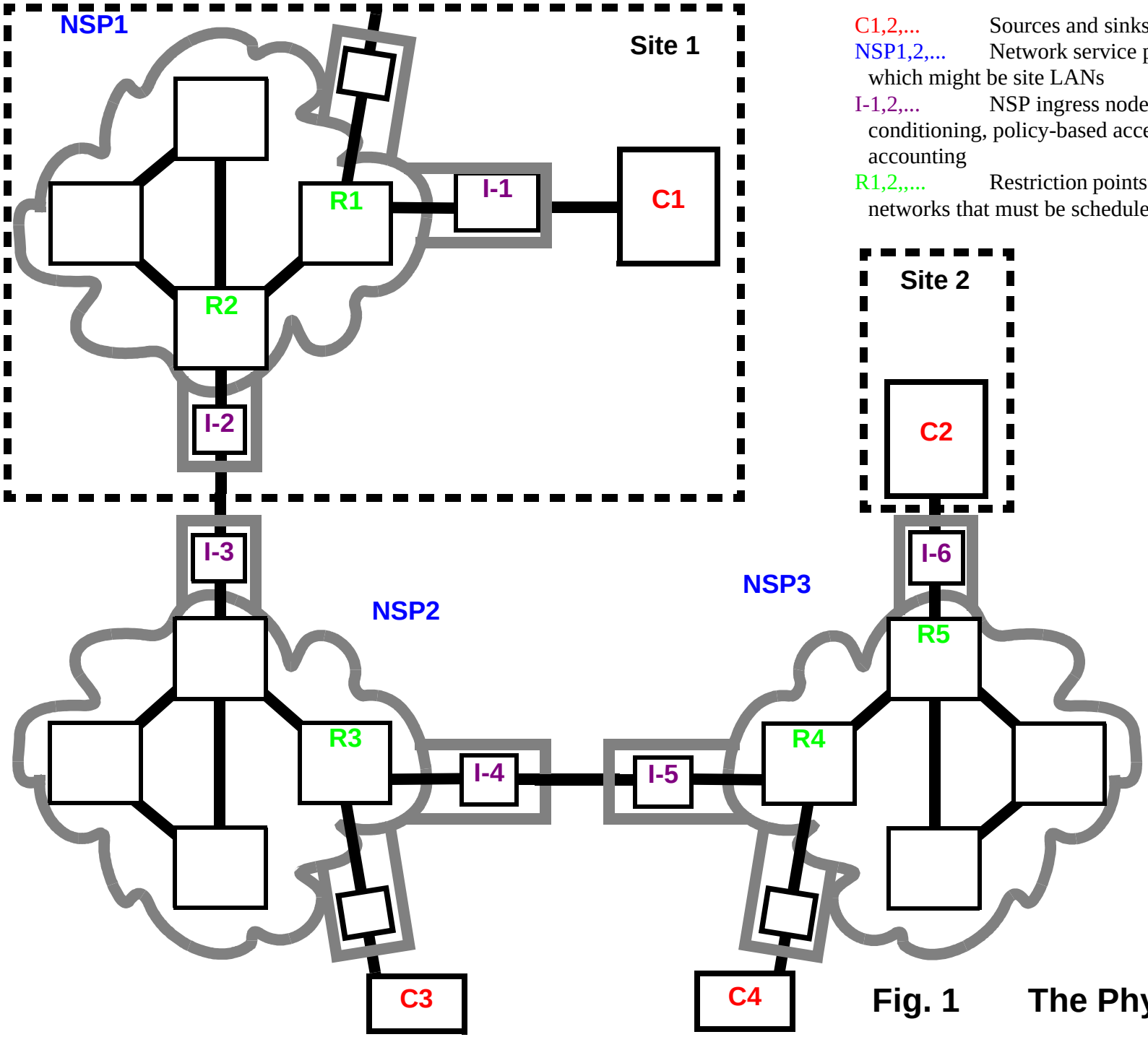
*This work is supported by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office, under contract DE-AC03-76SF00098 with the University of California and contract W-31-109-Eng-38 with the University of Chicago. The authors may be contacted at: qosdev@george.lbl.gov, itf@mcs.anl.gov, roy@mcs.anl.gov, sander@mcs.anl.gov. For more information see http://www.itg.lbl.gov/QoS/ .

Problem statement

Support for solving problems in *grid* (high-performance, distributed, internetworked) computing environments that require aggregating many resources, e.g., interconnect bandwidth.

The interconnect bandwidth will likely be from classes of premium (better than best effort) service. These service classes will be scarce compared to best-effort capacity, so NSPs can expect *theft of service* attacks: attempts to violate (circumvent or exceed the limits of) the policies governing access to the premium bandwidth.

A scheduling and reservation system vulnerable to fraud and theft is unlikely to be deployed by network administrators.



C1,2,... Sources and sinks

NSP1,2,... Network service providers, some of which might be site LANs

I-1,2,... NSP ingress nodes that provide traffic conditioning, policy-based access control, and accounting

R1,2,,... Restriction points in the interiors of the networks that must be scheduled

Fig. 1 The Physical Model

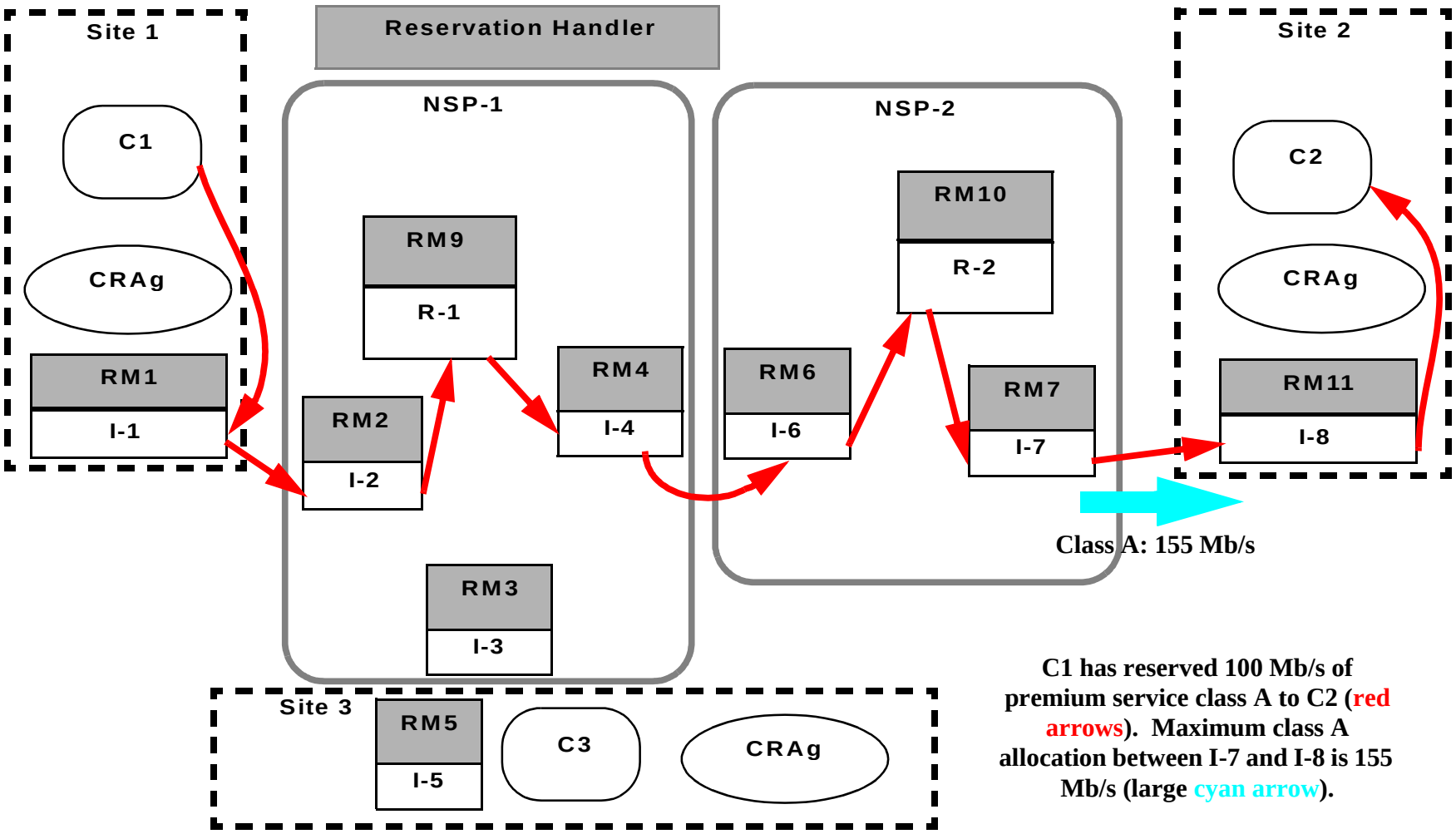
Design goals

- **Mechanism for secure, advance, end-to-end reservation of bandwidth**
  - **No design-imposed limit on the amount of bandwidth that may be reserved**
- **Allow control (policy-based access, scheduling) of premium service within a domain to remain with the network service provider**
- **Resist theft-of-service attacks**
- **Support (simple) negotiation**

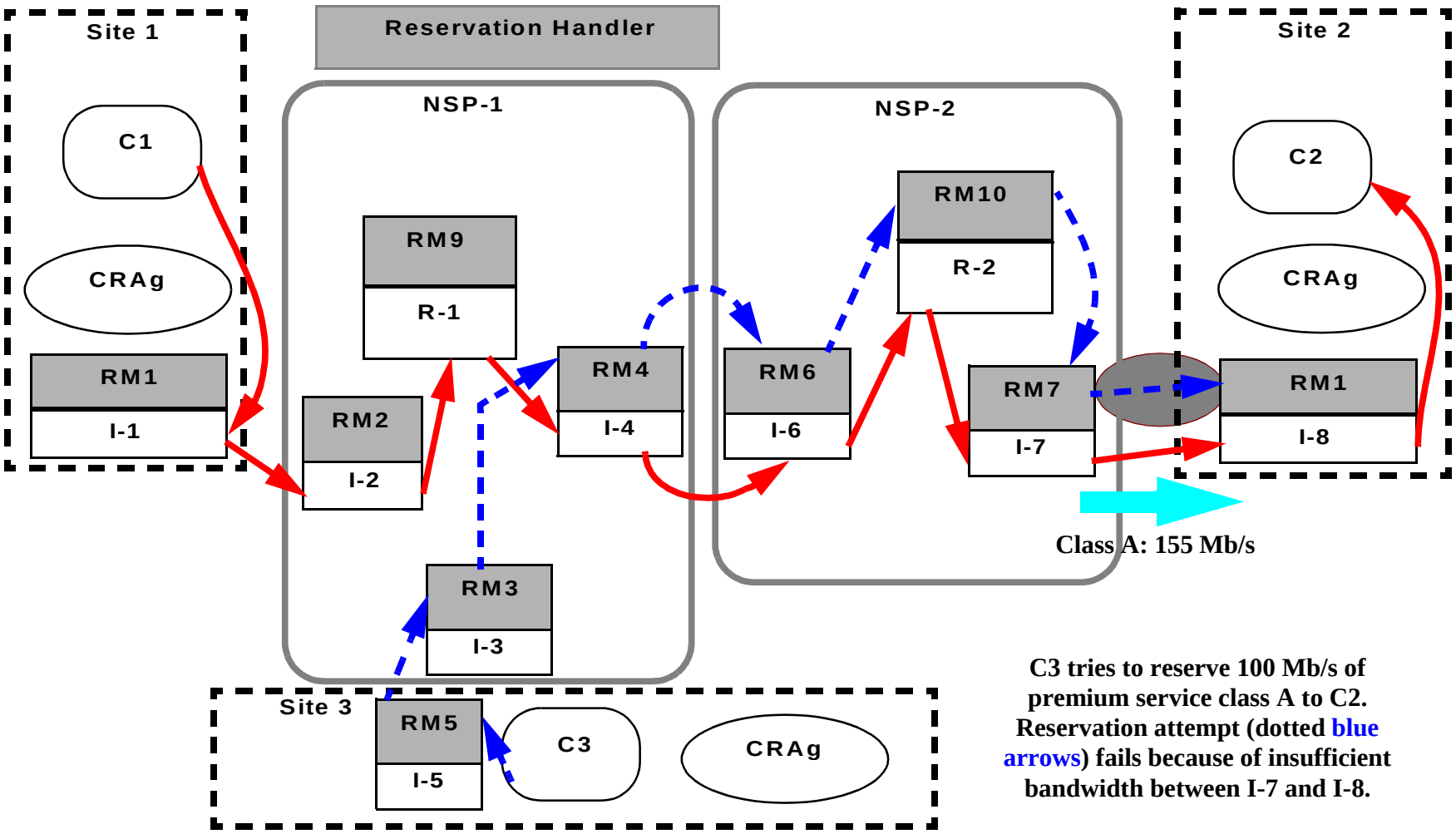
  (I.e., a way to query for available priority bandwidth within some given range of bandwidth and time periods to allow a broker function to choose the best slot.)
- **Provide end-to-end reservation path discovery**

Design goals

- **Support for accounting for use of priority service**
  - **however, the accounting's purposes and consequences are policy (political) issues not dictated by the system**
- **Preemptive reservation cancellation mechanism with notification**
  (in case of unforeseen path change invalidating reservation)
- **Claiming of reservation, at flow start-up, must be lightweight without sacrificing service guarantees**



C1 has reserved 100 Mb/s of premium service class A to C2 (red arrows). Maximum class A allocation between I-7 and I-8 is 155 Mb/s (large cyan arrow).



C3 tries to reserve 100 Mb/s of premium service class A to C2. Reservation attempt (dotted blue arrows) fails because of insufficient bandwidth between I-7 and I-8.

Elements of the architecture

- **Resource manager**
- **Request preprocessor**
- **Broker**
- **Client-side reservation agent**
- **Grid resources: compute, network, storage, scientific instruments, ... (not covered here)**
- **(and an application or two, also not covered here...)**

Elements of the architecture

- ***Resource manager (RM)***
  - **Provides "generic" services to resource(s):**
    - **authorization/access control (Akenti) under control of NSP**
    - **"wraps" advance reservation (including modification and cancellation) and resource manipulation functions**
    - **ingress ("first-hop") RM performs lightweight authorization checking at claim time**
    - **provides query interface for simple negotiation by higher-level broker**
    - **creates digitally signed reservation guarantees (*reservation tokens*) to ensure (1) authenticity, and (2) non-repudiation of reservations**
    - **reservation tokens serve as accounting records**

Elements of the architecture

- ***Request preprocessor***
  - **creates end-to-end network reservation**
    - **contacts each RM in the *reservation path* while each RM returns “success” token (otherwise, terminates reservation process)**
    - **allows higher-level components like Globus co-allocator and brokers to treat the network as a single component rather than a collection of resources each requiring a separate reservation**
  - **may modify/reformat requests before handing them to RM**

Elements of the architecture

- ***Broker***
  - **general resource negotiator and aggregator**
  - **responsible for coordinating reservations on all of the resources (bandwidth on network paths, CPUs on multiple systems, etc.) required to accomplish a task**
  - **has to be able to query resources for available slots so that it may find a common time interval over all required resources**

Elements of the architecture

- ***Client-side reservation agent (CRAg)***
  - **Encapsulates behavior/functionality common to all applications requiring reserved premium network bandwidth**
  - **Collects reservation path information for request preprocessor**
  - **Participates in authentication challenge with resource manager's security module to prove reservation requestor's identity**

Status

- **RM design complete**
  - **design has been updated and modified via extensive, ongoing discussions with ESNet, Globus, Internet2 QBone bandwidth broker developers; implementation underway**
  - **looking at NGI application requirements to ensure that the design and implementation are adequate to meet their needs**